Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US05/007655

International filing date: 09 March 2005 (09.03.2005)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/551,610
Filing date: 09 March 2004 (09.03.2004)

Date of receipt at the International Bureau: 20 April 2005 (20.04.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

1304411

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

April 04, 2005

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: ***60/551,610***
FILING DATE: ***March 09, 2004***
RELATED PCT APPLICATION NUMBER: ***PCT/US05/07655***

UNITED STATES PATENT AND TRADEMARK OFFICE · DEPARTMENT OF COMMERCE

Certified by

Jon W. Dudas

Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

16523 U.S. PTO
030904

Please type a plus sign (+) inside this box → +

PTO/SB/16 (5-03)
Approved for use through 4/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

22151 U.S. PTO
60/551610
030904

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
William L.	**Gaddy**	**P.O. Box 427 Urbana, IL 61803 - USA**
Chang	**Feng**	**4 Grant Court Metuchen, NJ 08840 - China**

☒ *Additional inventors are being named on the____ separately numbered sheets attached hereto*

TITLE OF THE INVENTION (280 characters max)

SYSTEM AND METHOD FOR PEER-TO-PEER CONNECTION OF CLIENTS BEHIND SYMMETRIC FIREWALLS

Direct all correspondence to: **CORRESPONDENCE ADDRESS**

☐ Customer Number ______ → *Place Customer Number Bar Code Label here*

OR *Type Customer Number here*

☒ Firm *or* Individual Name	**Richard C. Woodbridge**				
Address	**Synnestvedt Lechner & Woodbridge LLP**				
Address	**P.O. Box 592**				
City	**Princeton**	State	**NJ**	ZIP	**08542**
Country	**US**	Telephone	**609-924-3773**	Fax	**609-924-1811**

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒ Specification *Number of Pages* **11** ☐ CD(s), Number ____

☒ Drawing(s) *Number of Sheets* **10** ☐ Other (specify) ____

☐ Application Data Sheet. See 37 CFR 1.76

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

☒ Applicant claims small entity status. See 37 CFR 1.27.

☒ A check or money order is enclosed to cover the filing fees

☒ The Director is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number **23-3040**

☐ Payment by credit card. Form PTO-2038 is attached.

FILING FEE AMOUNT ($): **$80.00**

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are: ______

Respectfully submitted,

SIGNATURE Roy Rosser.

Date 8/9/04

TYPED or PRINTED NAME **Roy Rosser**

REGISTRATION NO. *(if appropriate)* **53,533**

Docket Number: **5636-104P**

TELEPHONE **609-924-3773**

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

PROVISIONAL APPLICATION COVER SHEET
Additional Page

PTO/SB/16 (8-00)
Approved for use through 10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a c ollection of information unless it displays a valid OMB control number.

Docket Number	5636-104P	Type a plus sign (+) inside this box ⟶	+

INVENTOR(S)/APPLICANT(S)		
Given Name (first and middle [if any])	Family or Surname	Residence (City and either State or Foreign Country)
Timothy Michael	**Hingston**	**301 Park Avenue** **Highstown, NJ 08520 - USA**
Chidambaram	**Ramanathan**	**2901 Holland Drive** **Somerset, NJ 08873 - India**

Number 2 of 2

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

EXPRESS MAIL NUMBER EV 044 714 045 US
FILE NUMBER 5636-104P

"EXPRESS MAIL CERTIFICATE"

"Express Mail" Mailing Label Number: EV 044 714 045 US

Express Mail Corporate Account Number: X085783

Date of Deposit: 3-9-04

Title: SYSTEM AND METHOD FOR PEER-TO-PEER CONNECTION OF CLIENTS BEHIND SYMMETRIC FIREWALLS

Inventors: William L. Gaddy; Chang Feng; Timothy Michael Hingston; Chidambaram Ramanathan

Type of Documents:

1. Provisional Application Cover Sheet – 2 pages;
2. Provisional Patent Application – 11 pages
3. Check in the amount of $80.00
4. 10 Drawings
5. Verified Statement Declaring Small Entity – 2 pages
6. This "Express Mail" Certificate; and;
7. Acknowledgment Post Card.

I hereby certify that the enclosed documents are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the USPTO; Mail Stop Provisional; Commissioner of Patents; PO Box 1450; Alexandria, VA 22313-1450.

Roy Rosser, Ph.D., Patent Agent.
(Typed or printed name of person mailing paper or fee)

Roy Rosser.
(Signature of person mailing paper of fee)

TITLE: System and Method for Peer-to-Peer Connection of Clients behind Symmetric Firewalls

INVENTORS: William L. Gaddy; Chang Feng; Timothy Michael Hingston; Chidambaram Ramanathan

FIELD OF THE INVENTION

[0001] The present invention relates to systems and methods of peer-to-peer communication and particularly to systems and methods of establishing direct Internet Protocol (IP) packet-based datagram communication between clients that are behind any combination of firewalls/Network Address Translation (NAT)s that allow outgoing Universal Data Packet (UDP) traffic, without port-forwarding, and without relaying or proxy services.

BACKGROUND OF THE INVENTION

[0002] In certain types of services over IP packet-switched networks, it is highly desirable to allow peer-to-peer communication between end-users. It is also highly desirable for any given method to allow as many as possible combinations of clients to communicate with each other. The lack of a successful method to accomplish this is a major reason behind the lack of pervasive deployment of services such as video conferencing.

[0003] Video is characterized by large bandwidth requirements for each direction of communication -- and it does not take many concurrent connections to overwhelm a typical circuit. It is therefore very desirable to avoid concentrations of this type of traffic at bottlenecks where physical or simple monetary constraints prevent the successful forwarding of essentially unlimited volumes of traffic.

[0004] Further, it is very desirable to minimize the time and efforts of specialized personnel required to support a given method. Some methods present problems of cost due to maintenance, setup, or security concerns.

[0005] There are several existing methods to traverse firewalls, in order to allow peer-to-peer modality for voice and video, including UDP Hole Punching (Internet Engineering Task Force (IETF) MidCom Working Group, P2PNAT (Peer-2-Peer Network Address Translation) Draft 2), and UPnP (Universal Plug and Play, Microsoft, et. al) but all of these have the problem

in that the range of firewalls and combinations thereof that support peer-to-peer connectivity when using them are limited.

[0006] UPnP

[0007] Simply stated, any client behind a suitably-configured UPnP firewall/NAT can map ports directly to the outside internet, and thereby look to any other outside client as a server for those ports. Most firewalls, regardless of type, are configured to allow client/server connections. However, the flaw of this protocol is that it has only been embraced by consumer device manufacturers. There are, for example, no enterprise-class firewalls with UPnP support. Therefore, UPnP does not solve any problems for enterprise-to-enterprise connectivity, and only works in the cases where one or both peers are behind firewalls/NATs that support it.

[0008] UDP Hole Punching

[0009] UDP Hole Punching is more limiting. As envisaged by the IETF MidCom working group, both firewalls/NAts must be of a Cone-UDP type (this is generally specific to low-end consumer stateless firewalls). The probabilities of actual circumstance of these cases are multiplicative, and unfortunately, therefore, relatively rare—especially in the enterprise-to-consumer and enterprise-to-enterprise cases.

[0010] Other methods

[0011] If one wants to enable video communication between any two arbitrary clients where both are behind symmetric firewalls (generally, enterprise-to-enterprise), there are three choices, all of which either engender the aforementioned concentrations of traffic and the expenses accruing thereto, or that require specialized installation, configuration, and/or active management and monitoring by qualified personnel of proprietary proxy/relay solutions for at least one of the peers' internal networks.

[0012] These three choices are:

[0013] 1. To require at least one of the clients to be behind a firewall that has built-in or installed capability to support dynamic port-forwarding according to a common signaling and call origination protocol, such that said firewall can ensure that the used ports are forwarded in such a way that the client behind the forwarded firewall appears as a server to the client behind the other firewall, or;

[0014] 2. To require proxy/relay services located in the DMZ of one of the clients' firewalls, to allow communication between a peer behind the proxied firewall and one outside -- where, again, the client behind the proxied service looks like a server to the other client, or;

[0015] 3. To locate a proxy/relay service behind a known single address or group of addresses that is outside of both peer's firewalls to relay the traffic, wherein both clients are communicating with a common server – the relay.

[0016] The first choice is exemplified by H.323 and SIP—both are well-known connection and signaling protocols for establishing peer-to-peer connections over IP networks. They are supported by many enterprise firewalls, but not all. They also are supported by hardly any mass-market consumer hardware and software firewalls. Because these protocols use many and/or arbitrary TCP and UDP ports, these protocols are difficult to trace, more difficult to analyze and monitor, and many firewall administrators simply turn these protocol capabilities off in the firewalls that do have native support for it, rather than be tasked with monitoring and managing them. Furthermore, discoveries about security holes in the reference implementation of H.323 will undoubtedly result in this protocol being disabled by many administrators. In general, this method could work if there was a protocol that met the requirements for security, manageability, pervasiveness and adoption -- but this is not the case with H.323 and SIP and no protocols are currently on the standards-track that satisfy all of the foregoing requirements.

[0017] The second choice has become the preferred method of managing peer-to-peer video services in the enterprise -- however, the costs accruing to it are asymmetric. Since it requires at least one client to be behind a firewall whose administrator has provided a video relay service in the DMZ (and at the costs associated with it), an all-too-defensible position from an IT Management perspective is that if video services are so necessary between "us" and "them", why don't "they" absorb the cost of installing and maintaining a proxy/relay service? A common consequence is that no one ends up absorbing this expense.

[0018] The third choice is a natural consequence of the drawbacks of the first two: there are presently no interoperable, standards-based solutions which require less than significant expense that allow any two given clients behind any two symmetric firewalls to communicate with each other . If one could provide a third party relay service, and absolve individual end-user firewall administrators of this task, it would vastly simplify the administrators' overall architecture, equalize costs among end users, and provide a common service provider point.

Unfortunately, the common point(s) are the root of the failure for this method to provide an cost-effective and scalable solution to video connectivity. In order to support such a solution for 100,000 concurrent two-way video conferences, each using (conservatively) 200 kBit each way, a central relay service must support 40,000 MBit circuit connectivity (4000 T1 circuits). For each additional user, another 400 kBit of capability must be added. Clearly, this is prohibitively expensive and does not scale well at all.

[0019] There appear to be no existing systems that can, at once, solve the stated problems of all of the above five methods (or combinations thereof) that prevent wide-spread adoption and usage by end-users, by simultaneously allowing true peer-to-peer, unproxied/unrelayed connections between all of the following:

Clients behind Cone or UPnP Firewalls/NATs to clients behind same;

Clients behind Cone or UPnP Firewalls/NAT's to clients on routable addresses;

Clients behind Cone or UPnP Firewalls/NAT's to clients behind Symmetric Firewalls/NATs; and

Clients behind Symmetric Firewalls/NATs to clients behind routable addresses;

Clients behind Symmetric Firewalls/NATs to clients behind Symmetric Firewalls/NATs.

SUMMARY OF THE INVENTION

[0020] An object of the current invention is to allow peer-to-peer connectivity between clients, regardless of the type of firewall/NAT each is behind, whether Cone (Figure 1), Port-Restricted Cone (Figure 2), Symmetric (Figure 3), or any combinations thereof, without specific protocol support, installation of per-client server/services, or configuration of one or both clients' firewalls/NATs.

[0021] A further object of the current invention is to allow peer-to-peer connectivity between multiply-NAT-ted clients, some of said NATs being symmetric in nature, under limited circumstances, that was otherwise impossible with any other method or combinations of methods.

[0022] To achieve the first object, a method of establishing peer-to-peer connectivity between clients behind symmetric or cone firewalls/NATs must include discovering what the proper tuple (source/destination port, and source/destination address combination) is required to

allow the client's firewall to forward packets to the client. In addition, the symmetric port translation behavior of firewalls can be further characterized as Symmetric Second Priority PAT (Figure 4A), and Symmetric Pure PAT (Figure 4b). Ultimately the calling client wants to establish two-way communication with a called client, and to do so each much know what port was assigned to the address combination on both of the clients' NAT/PATs. The problem inherent with achieving this is illustrated in Figure 5.

[0023] A first step to accomplish the first object is to obtain each client's publicly routable address and an example of a publicly routable, masqueraded port by contacting a discovery server. Since each separate destination server address (and, ultimately the called client's destination address) results in a different port mapping for Symmetric NAT/PATs, a second request to a second discovery server is indicated. This also simplifies the cases such as in Figure 4a where in a very under-utilized NAT/PAT the port address translation will give a direct port mapping to the first internal user of a given port, but a masqueraded port for subsequent address contacts. It is thus ensured that the second and subsequent addressed requests will use masqueraded ports.

[0024] The calling client retrieves this information from the discovery servers, and sends the second tuple (combination of source/destination port, source/destination address) to the called client via a well-known, open, and agreed server, as in Figure 5.

[0025] In response, the called client does the same for itself, and responds to the calling client with its second tuple. The called client also begins sending UDP packets to the reported source address and source port of the calling client. If the calling client is a Cone NAT, these packets will be delivered. If the calling client is behind a Symmetric NAT, they will not (as in Figure 5).

[0026] In the meantime, when the calling client receives the called client's tuple, it, too will begin to send UDP packets to the called client's reported source address and source port. If the called client is behind a Cone NAT, these packets will be delivered. If the called client is behind a Symmetric NAT, they will not (as in Figure 5).

[0027] Once a client receives an inbound packet, it knows what the proper destination port of its peer is, regardless of what type of firewall/NAT the other client is behind.

[0028] If one of the clients happens to be behind a Cone NAT, the first few attempts at sending to the original destination port will succeed. When the firewall forwards the packet, the

client will receive it, take note of the inbound packet's source port, and will then know to send all traffic to that destination port. In addition, the client will send a success packet to indicate to the other client that it can stop sending discovery packets. Up to this stage, the process may be much like a normal UDP Hole Punch combined with a connection-reversal. The next part of the process departs significantly from normal UDP Hole Punch methods..

[0029] In the case where both clients are behind symmetric NATs, neither client will receive UDP packets.

[0030] When a certain period of time has elapsed before a client has received one of these UDP packets, the client will begin to send its packets not to a single destination port, but to an entire range of ports ("Shotgun"). Most firewall/NATs that perform port masquerading use a simple algorithm (usually simple addition) to assign ports to clients sending UDP requests. A wide enough range will likely "find" the masqueraded port of the other peer by brute force. When the firewall forwards the packet, the client will receive it, take note of the inbound packet's source port, and will then know to send all traffic to that destination port. If both clients are behind symmetric firewalls, they both will send this series of UDP packets to "find" the active port, and both clients will discover the active destination port of their peer. Figure 6 is a full traffic and tuple diagram of this process, including the important firewall state table tuples at each point of the exchange.

[0031] The figure omits the second discovery server contact for brevity. In addition, the "Shotgun" width described in the figure is limited to the range of the original port through the original port plus 8. The preferred embodiment uses a much wider range (minus 16 through plus 32), but the full range is not included in the figure for brevity.

[0032] When a client gets a positive indication of an incoming packet, it sends a success packet response to the sender to indicate that it can stop sending discovery packets. This always succeeds, because the client sending the response now always knows what destination port to send to. Figure 7 depicts a flowchart of the entire protocol exchange as described.

[0033] Subsequently, all payload is sent from a given client using this identified port.

[0034] To achieve the second object of the invention, both clients use UPnP support, if available, as a first step to directly map ports, thus ensuring a connection reversal. The further ability to match source port and masqueraded destination ports offers the ability to communicate with symmetric firewalls that have been manually configured to not allow outgoing UDP

requests on the dynamic port range. Figure 8 depicts a flowchart of the entire protocol exchange including the UPnP steps.

BRIEF DESCRIPTION OF THE DRAWINGS

[0035] FIG. 1 shows a representation of requests and responses in a system in which a client is behind a Cone NAT/PAT.

[0036] FIG. 2 shows a representation of requests and responses in a system in which a client is behind a Port-Restricted Cone NAT/PAT.

[0037] FIG. 3 shows a representation of requests and responses in a system in which a client is behind a Symmetric NAT/PAT.

[0038] FIG. 4a shows a representation of requests and responses in a system in which a client is behind a second-priority masquerading NAT/PAT.

[0039] FIG. 4b shows a representation of requests and responses in a system in which a client is behind a pure masquerading NAT/PAT.

[0040] FIG. 5 shows a representation of the initial discovery requests and responses in a connection reversal failure between clients behind symmetric NAT's.

[0041] FIG. 5b shows a representation of a connection reversal failure between clients behind symmetric NAT's.

[0042] FIG. 6 shows a representation of an initial stage of a shotgun exchange between clients behind symmetric NAT/PATS's.

[0043] FIG. 6b shows a representation of a later stage of a shotgun exchange between clients behind symmetric NAT/PATS's.

[0044] FIG. 7 shows a flowchart of discovery, message exchange and the shotgun process.

[0045] FIG. 8 shows a flowchart of discovery, message exchange and the shotgun process using UPnP.

DETAILED DESCRIPTION

[0046] The present invention is The preferred embodiment of the method disclosed comprises:

[0047]

[0048] Two or more discovery servers are situated at different addresses, each listening at a series of well-known UDP ports, each of which will respond to well-formed requests from clients with a response containing the requesting client's public address and public port; and two clients who will execute the following steps of the method, in order:

[0049] Calling client determines if the local NAT, if present, supports UPnP

[0050] Calling client determines if the local NAT, if present, supports UPnP client-activated port forwarding

[0051] If affirmative 1 and 2, calling client attempts to map the source port to the destination port identically and directly across the NAT via UPnP

[0052] The calling client retrieves its private address, private source port, public address, public source port, and public destination port tuple by contacting and receiving response from a first discovery server at a first address via a well-known source and destination port (DUDP_START request, DUDP_PUBINFO response).

[0053] The calling client retrieves its private address, public address, private destination port, and public destination port tuple by contacting and receiving response from a second discovery server at a second address via the same well-known source and destination port as in 1 (DUDP_START request, DUDP_PUBINFO response).

[0054] The calling client will sends the contents of its received second tuple plus the differential of the first discovery-reported source port and second discovery-reported source port to the called client via an established, mutually agreed-upon server for this purpose (MESSAGE_CONTROL)

[0055] If the called client is not willing to receive calls from the sender, an abort is signaled to the sender and the process stops.

[0056] If the called client is willing to receive calls from the sender, the called client determines if the local NAT, if present, supports UPnP

[0057] The called client determines if the local NAT, if present, supports UPnP client-activated port forwarding

[0058] If affirmative 8 and 9, the called client attempts to map the source port to the destination port identically and directly across the NAT via UPnP

[0059] The called client will retrieve the calling client's tuple (MESSAGE_CONTROL), and its own source address, public address, source port, and destination port tuple by contacting and receiving response from a first discovery server via a well-known source and destination port. (DUDP_START request, DUDP_PUBINFO response)

[0060] The called client will retrieve its source address, public address, source port, and destination port tuple by contacting and receiving response from a second discovery server at a second address via the same well-known source and destination port as in 11. (DUDP_START request, DUDP_PUBINFO response).

[0061] The called client will send the contents of its received second tuple, plus the differential of the first discovery-reported source port and second discovery-reported source port, plus any desired modifications to the calling client's tuple, to the calling client via an established, mutually agreed-upon server for this purpose.

[0062] The called client will then begin a periodic send of UDP packets (DUDP_ACK) to the calling client's address and source port according to the tuple reported to it by the caller's MESSAGE_CONTROL when in good receipt..

[0063] The calling client, upon good receipt of a tuple response (MESSAGE_CONTROL) from the called client, will then begin a periodic send of UDP packets (DUDP_ACK) to the called client's address and source port according to the tuple reported to it by the called client's MESSAGE_CONTROL.

[0064] If the calling client receives a DUDP ACK, it will take note of the source port identified in the IP header of said packet, and use it for subsequent outgoing DUDP_ACK packets, mark this port for further payload traffic, and also send a DUDP_ACK2 packet to this destination port. If no DUDP_ACK is received within a certain period of time, a series of DUDP_ACK packets, each with a destination port within a range beyond and contiguous to a predicted value extrapolated by the differential reported in 9, is sent periodically instead of a single packet to a single destination port. Subsequent, repeated transmissions of this series may move the port range window with each iteration.

[0065] If the called client receives a DUDP ACK, it will take note of the source port identified in the IP header of said packet, and use it for subsequent outgoing DUDP_ACK packets, mark this port further payload traffic, and also send a DUDP_ACK2 packet to this destination port. If no DUDP_ACK is received within a certain period of time, a series of

DUDP_ACK packets, each with a destination port within a range beyond and contiguous to a predicted value extrapolated by the differential reported in 6, is sent periodically instead of a single packet to a single destination port. Subsequent, repeated transmissions of this series may move the port range window with each iteration.

[0066] If the calling client either times out, or receives a DUDP_ACK2, it assumes that it has a properly marked destination port, using the reported called client's reported tuple source port as a destination port failover value.

[0067] If the called client either times out, or receives a DUDP_ACK2, it assumes that it has a properly marked destination port, using the reported calling client's reported tuple source port as a destination port failover value.

[0068] When the calling client has a properly marked destination port, it will begin to send payload data to this port to the called client.

[0069] When the called client has a properly marked destination port, it will begin to send payload data to this port to the calling client.

[0070] The foregoing embodiment is strictly exemplary in nature and is not to be construed as limiting the present invention. Many variations and modifications of the invention will be readily apparent to those skilled in the art.

ABSTRACT

A system and method for establishing and maintaining two-way peer-to-peer internet communication between clients who are behind symmetric firewalls/NATs is presented. The method uses several third-party address-and-port discovery servers to ascertain the nature and port-mapping metrics of a given client's firewall/NAT. A systematic, multiple UDP Hole Punch method is employed for ports within a predicted range, and the source port of the first successful forwarding of an inbound packet is used by the client for subsequent outgoing traffic. This occurs symmetrically, thus ensuring that both clients' firewalls receive packets for which the source/destination ports and source/destination addresses fully-tuple-match with a previous client request originating from within the protected network, and therefore forwards packets to the respective clients successfully (peer-to-peer). In addition, the method allows monitoring, management, and prevention of connections by interested firewall/NAT administrators

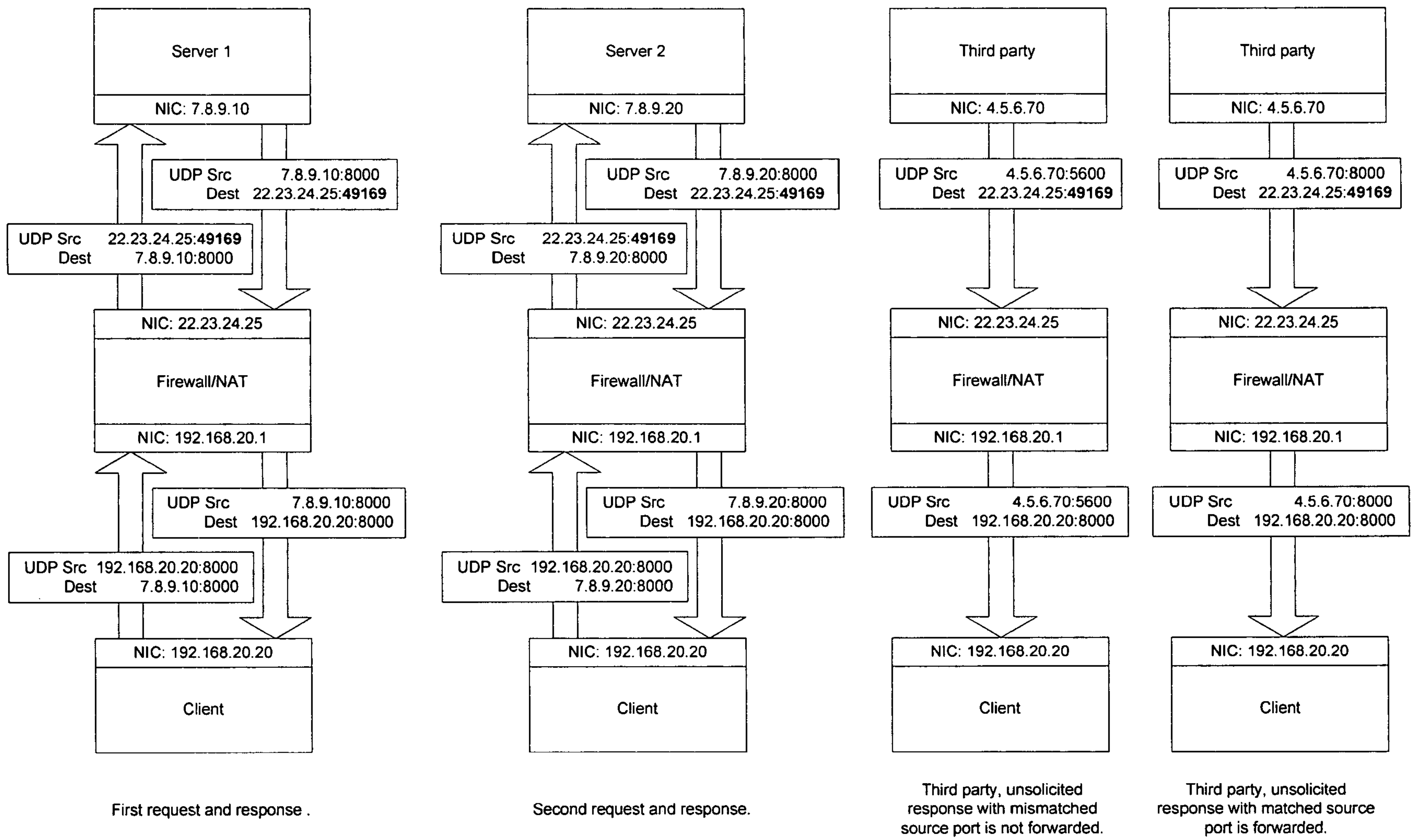

Figure 1: Client behind a Cone NAT/PAT. All requests with a given port (8000) are masqueraded. The next, and subsequent requests made to different destination addresses with that port are masqueraded to the same port. In addition, unsolicited responses from others addresses are forwarded, regardless of source port.

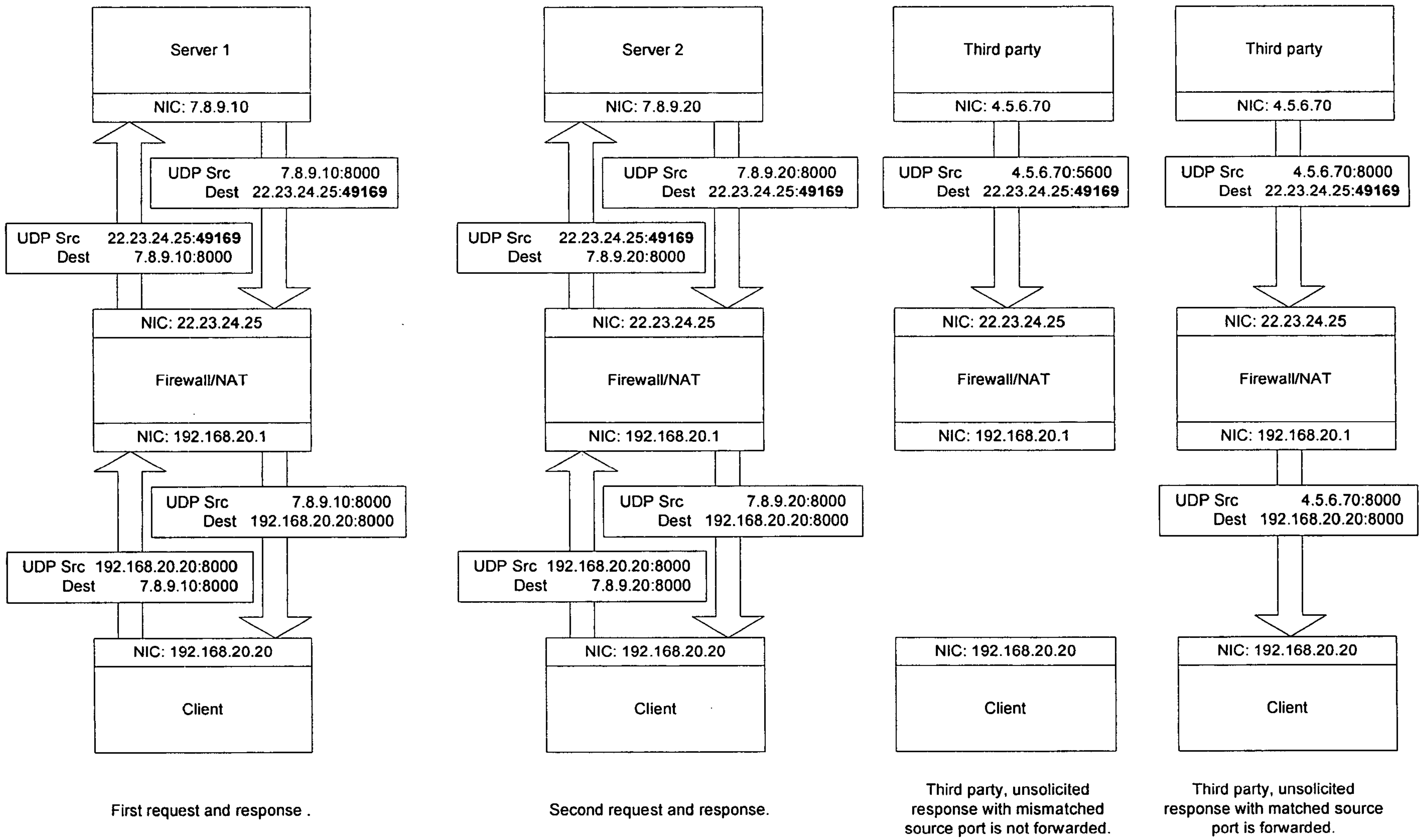

Figure 2: Client behind a Port-Restricted Cone NAT/PAT. All requests with a given port (8000) are masqueraded. The next, and subsequent requests made to different destination addresses with that port are masqueraded to the same port. In addition, unsolicited responses from others addresses are forwarded, so long as the source port matches the destination port of the original request.,

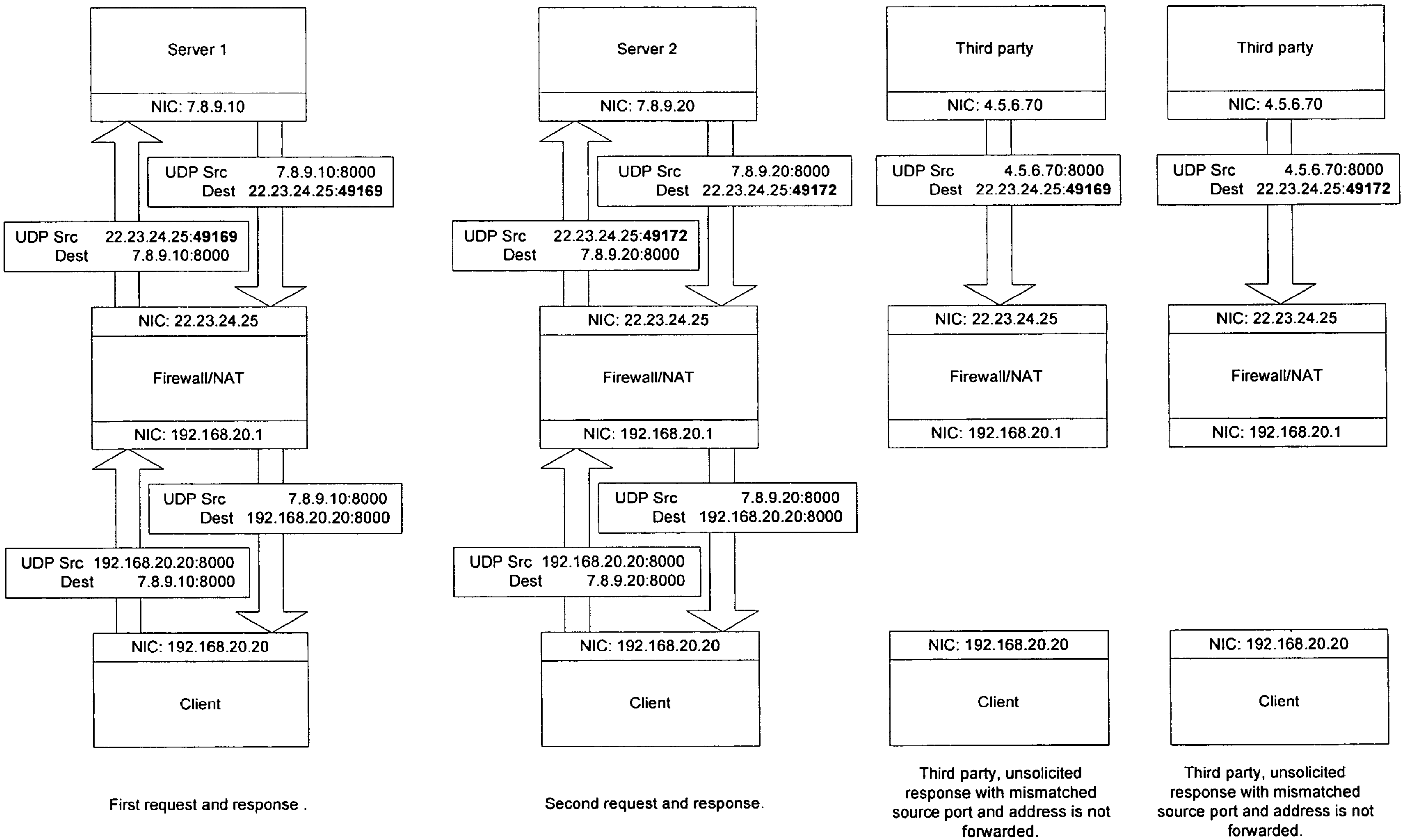

Figure 3: Client behind a Symmetric NAT/PAT. All requests with a given port (8000) are masqueraded. The next, and subsequent requests made to different destination addresses with that port are masqueraded to different ports. In addition, unsolicited responses from others addresses are not forwarded, because the source addresses do not match the destination of the original request., Most symmetric firewalls also require the source port to match the destination port of the original request (full tuple match).

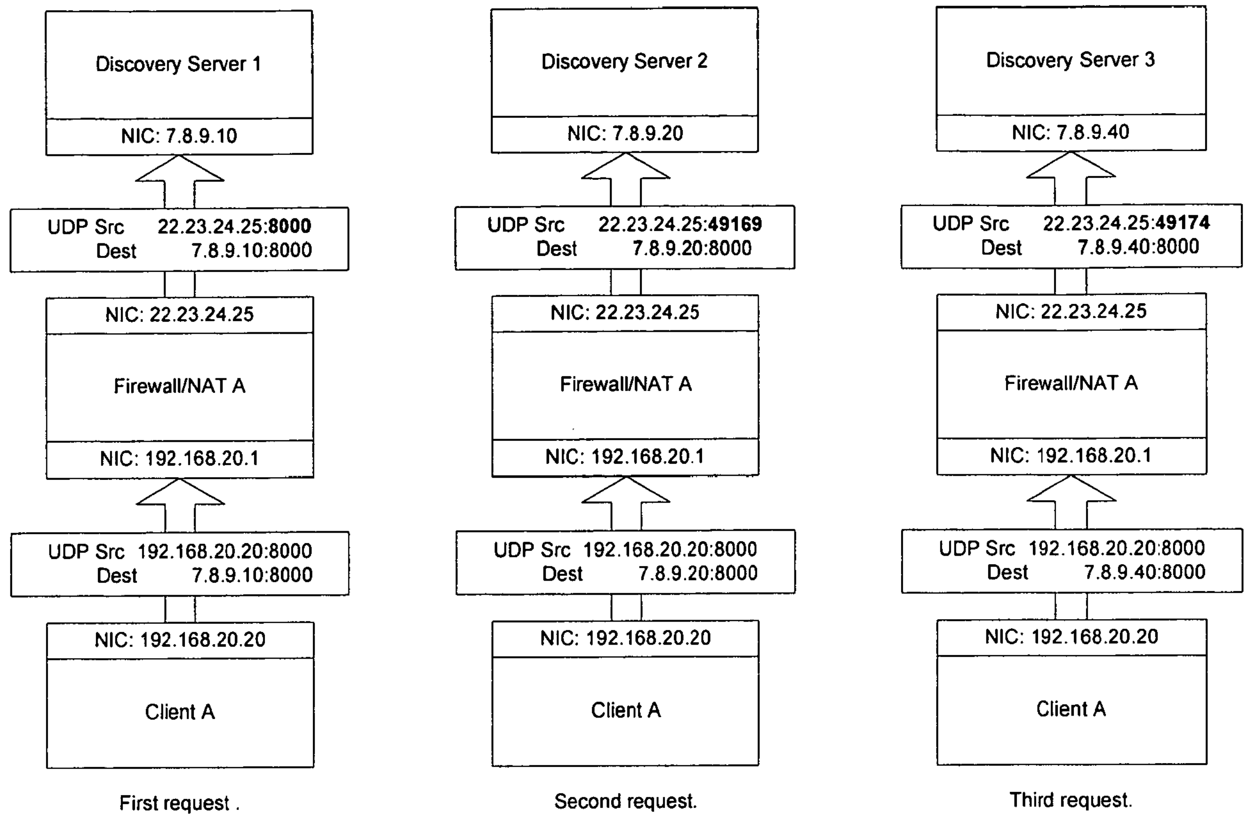

Figure 4a: Client behind a second-priority masqerading NAT/PAT. The first request with a given port (8000) is not masqueraded. The next, and subsequent requests made to different destination addresses with that port before the first one expires are masqueraded.

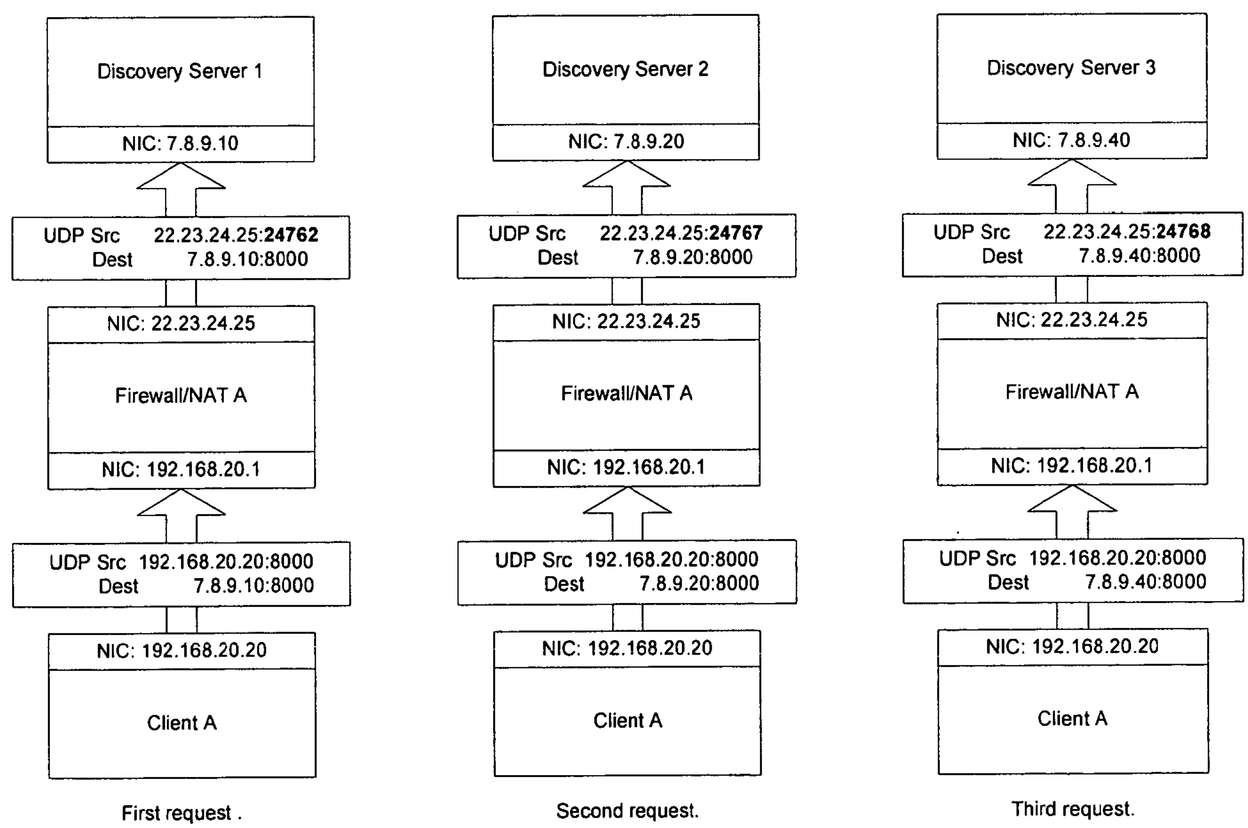

Figure 4b: Client behind a pure masqerading NAT/PAT. All requests with a given port (8000) are masqueraded. The masqueraded port changes for each destination address.

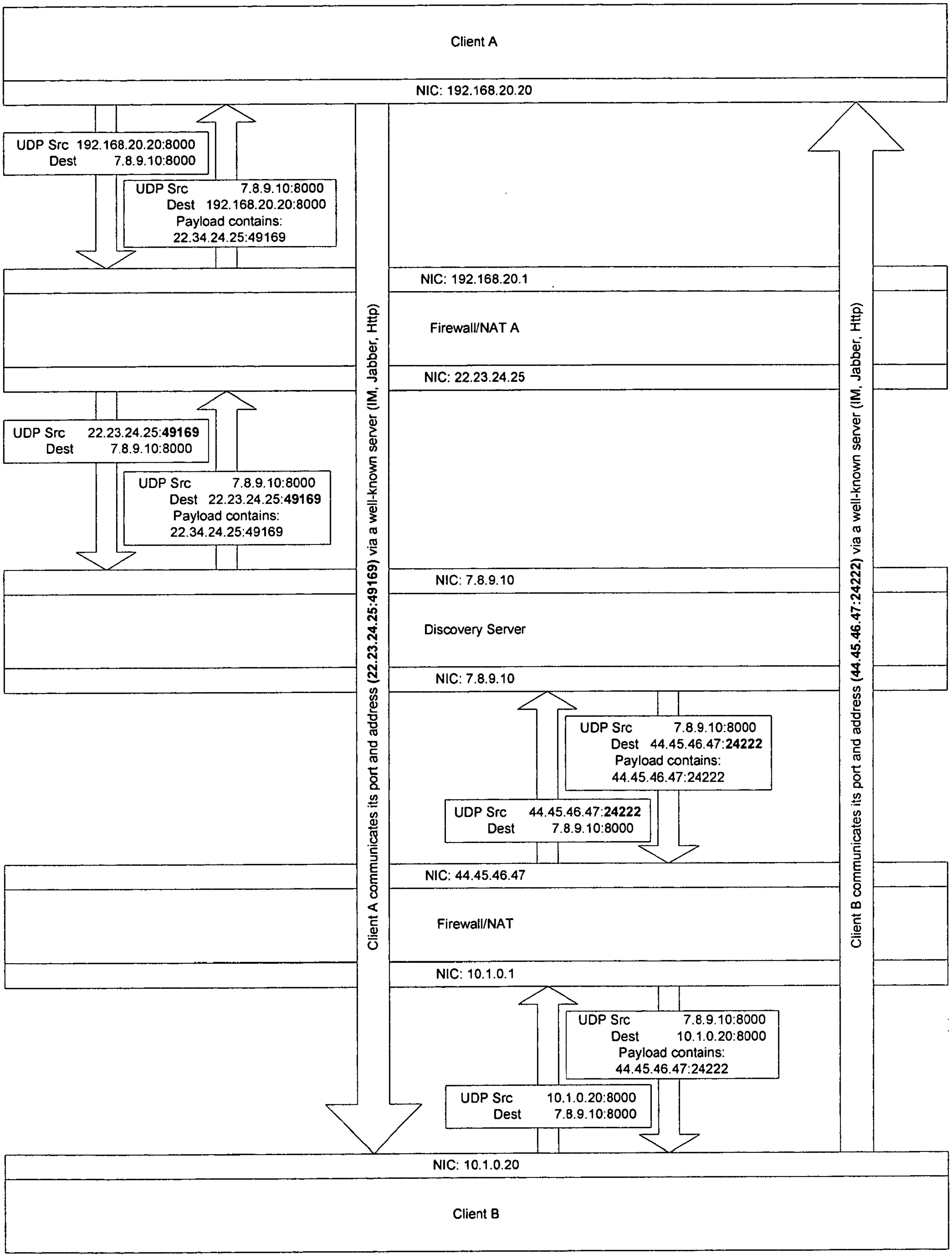

Figure 5: Connection reversal failure between clients behind symmetric NATs (left overleaf) Shows the initial discovery server requests to get masqueraded ports and external routable addresses, and the exchange of same between the clients.

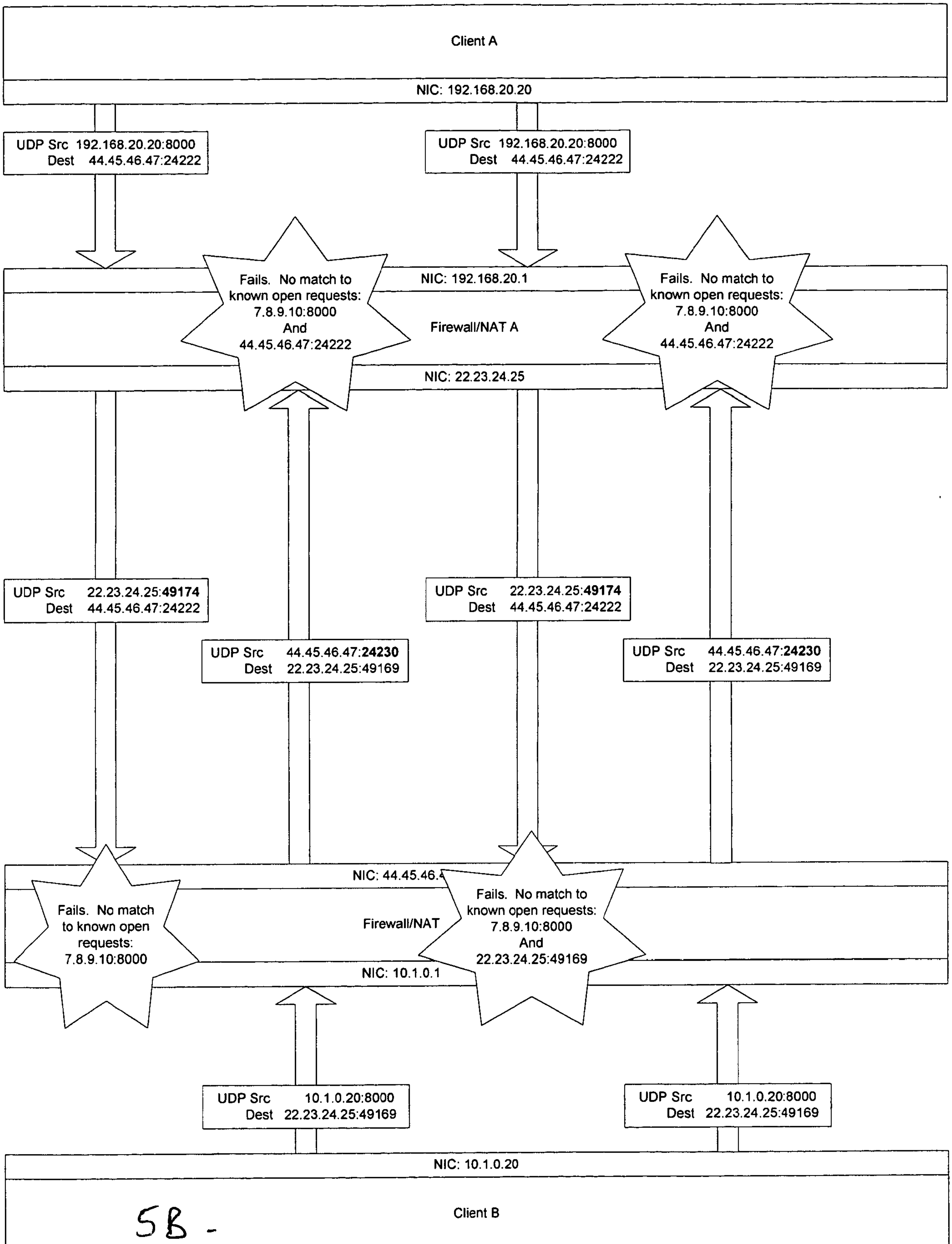

Figure 5: Connection reversal failure between clients behind symmetric NATs (right overleaf) Shows the failed attempt to use the discovered ports to communicate directly between the clients.

```
Brute-force Shotgun (no second server port prediction)
Non-UPnP Symmetric NAT (A) and Non-UPnP Symmetric Nat (B) [CheckPoint to CheckPoint]*
Firewalls assume:
        1. NAT A = one state table entry per source port, source address, destination port, destination address combination
        2. NAT A = only re-map SNAT ports for different destination address combinations
        3. NAT B = one state table entry per source port, source address, destination port, destination address combination
        4. NAT B = only re-map SNAT ports for different destination address combinations
(*NAT A behaves like CheckPoint in default configuration, NAT B behaves like CheckPoint in default configuration)

Shotgun spread = port+0 though port+8 (typical usage is port-16 to through port+32)

T+ 0.0s
                         Client A (192.168.10.140)      NAT/FW A(192.168.10.1, 12.181.128.1)      DS01 (7.8.9.10)     NAT/FW B(24.181.10.1, 10.0.0.1)        Client B (10.0.0.150)
                         -------------------------      ------------------------------------      ---------------     -------------------------------        ---------------------
Both clients contact     S 192.168.10.140:5432  SNAT    S 12.181.128.1:24154                                          S 24.181.10.1:49153               SNAT  S 10.0.0.150:5432
Disco server             D       7.8.9.10:5432   =>     D       7.8.9.10:5432             =>      Reflection-   <=    D       7.8.9.10:5432              <=   D   7.8.9.10:5432
(DUDP_START).

Both clients receive     S       7.8.9.10:5432  DNAT    S       7.8.9.10:5432                                         S       7.8.9.10:5432             DNAT  S   7.8.9.10:5432
Disco server resp.  D 192.168.10.140:5432        <=     D 12.181.128.1:24154              <=      Response.     =>    D 24.181.10.1:49153                =>   D 10.0.0.150:5432
(DUDP_PUBINFO).

Clients A and B          Client A knows Client B to be at 24.181.10.1:49153                                           Client B knows Client A to be at 12.181.128.1:24154
exchange their external
addresses and initial

external ports, out-of-

band.


T+ 0.5s                                                                                                               Tuple match. (Firewall)
Client A sends first     S 192.168.10.140:5432  SNAT    S 12.181.128.1:24157                                          S 12.181.128.1:24157
packets to punch hole    D    24.181.10.1:49153  =>     D  24.181.10.1:49153                            =>            D  24.181.10.1:49153        no match.
(DUDP_ACK1)
                         S 192.168.10.140:5432  SNAT    S 12.181.128.1:24157                                          S 12.181.128.1:24157
(SHOTGUN)                D    24.181.10.1:49154  =>     D  24.181.10.1:49154                            =>            D  24.181.10.1:49154        no match.
                         S 192.168.10.140:5432  SNAT    S 12.181.128.1:24157                                          S 12.181.128.1:24157
(SHOTGUN)                D    24.181.10.1:49155  =>     D  24.181.10.1:49155                            =>            D  24.181.10.1:49155        no match.
                         S 192.168.10.140:5432  SNAT    S 12.181.128.1:24157                                          S 12.181.128.1:24157
(SHOTGUN)                D    24.181.10.1:49156  =>     D  24.181.10.1:49156                            =>            D  24.181.10.1:49156        no match.
                         S 192.168.10.140:5432  SNAT    S 12.181.128.1:24157                                          S 12.181.128.1:24157
(SHOTGUN)                D    24.181.10.1:49157  =>     D  24.181.10.1:49157                            =>            D  24.181.10.1:49157        no match.
                         S 192.168.10.140:5432  SNAT    S 12.181.128.1:24157                                          S 12.181.128.1:24157
(SHOTGUN)                D    24.181.10.1:49158  =>     D  24.181.10.1:49158                            =>            D  24.181.10.1:49158        no match.
                         S 192.168.10.140:5432  SNAT    S 12.181.128.1:24157                                          S 12.181.128.1:24157
(SHOTGUN)                D    24.181.10.1:49159  =>     D  24.181.10.1:49159                            =>            D  24.181.10.1:49159        no match.
                         S 192.168.10.140:5432  SNAT    S 12.181.128.1:24157                                          S 12.181.128.1:24157
(SHOTGUN)                D    24.181.10.1:49160  =>     D  24.181.10.1:49160                            =>            D  24.181.10.1:49160        no match.
                         S 192.168.10.140:5432  SNAT    S 12.181.128.1:24157                                          S 12.181.128.1:24157
(SHOTGUN)                D    24.181.10.1:49161  =>     D  24.181.10.1:49161                            =>            D  24.181.10.1:49161        no match.

                                                Known open session are now:                                           FW/Nat drops packets on floor.
                                                S       7.8.9.10:5432   opened and                                    No Match for known open sessions:
                                                D 12.181.128.1:24154    responded
                                                                                                                      S       7.8.9.10:5432   opened and
                                                S  24.181.10.1:49153    opened, not                                   D 12.181.128.1:24154    responded
                                                D 12.181.128.1:24157    responded

                                                S  24.181.10.1:49154    opened, not
                                                D 12.181.128.1:24157    responded
                                                S  24.181.10.1:49155    opened, not
                                                D 12.181.128.1:24157    responded
                                                S  24.181.10.1:49156    opened, not
                                                D 12.181.128.1:24157    responded
                                                S  24.181.10.1:49157    opened, not
                                                D 12.181.128.1:24157    responded
                                                S  24.181.10.1:49158    opened, not
                                                D 12.181.128.1:24157    responded
                                                S  24.181.10.1:49159    opened, not
                                                D 12.181.128.1:24157    responded
                                                S  24.181.10.1:49160    opened, not
                                                D 12.181.128.1:24157    responded
                                                S  24.181.10.1:49161    opened, not
                                                D 12.181.128.1:24157    responded
```

Figure 6: Shotgun Exchange between Client behind Symmetric NAT/PATs, part 1 of 2.

T+ 0.56s							
Client B sends first packets to punch hole (DUDP_ACK)	no match.		Tuple match. (Firewall) S 24.181.10.1:49159 D 12.181.128.1:24154	⇦	S 24.181.10.1:49159 D 12.181.128.1:24154	SNAT ⇦	S 10.0.0.150:5432 D 12.181.128.1:24154
(SHOTGUN)	no match.		S 24.181.10.1:49159 D 12.181.128.1:24155	⇦	S 24.181.10.1:49159 D 12.181.128.1:24155	SNAT ⇦	S 10.0.0.150:5432 D 12.181.128.1:24155
(SHOTGUN)	no match.		S 24.181.10.1:49159 D 12.181.128.1:24156	⇦	S 24.181.10.1:49159 D 12.181.128.1:24156	SNAT ⇦	S 10.0.0.150:5432 D 12.181.128.1:24156
(SHOTGUN)	S 24.181.10.1:49159 D 192.168.10.140:5432 DUDP_ACK = use port #.	DNAT ⇦	S 24.181.10.1:49159 D 12.181.128.1:24157	⇦	S 24.181.10.1:49159 D 12.181.128.1:24157	SNAT ⇦	S 10.0.0.150:5432 D 12.181.128.1:24157
(SHOTGUN)	no match.		S 24.181.10.1:49159 D 12.181.128.1:24158	⇦	S 24.181.10.1:49159 D 12.181.128.1:24158	SNAT ⇦	S 10.0.0.150:5432 D 12.181.128.1:24158
(SHOTGUN)	no match.		S 24.181.10.1:49159 D 12.181.128.1:24159	⇦	S 24.181.10.1:49159 D 12.181.128.1:24159	SNAT ⇦	S 10.0.0.150:5432 D 12.181.128.1:24159
(SHOTGUN)	no match.		S 24.181.10.1:49159 D 12.181.128.1:24160	⇦	S 24.181.10.1:49159 D 12.181.128.1:24160	SNAT ⇦	S 10.0.0.150:5432 D 12.181.128.1:24160
(SHOTGUN)	no match.		S 24.181.10.1:49159 D 12.181.128.1:24161	⇦	S 24.181.10.1:49159 D 12.181.128.1:24161	SNAT ⇦	S 10.0.0.150:5432 D 12.181.128.1:24161
T+ 0.57s							
Client A sends ACK2 Packet (DUDP_ACK2)	S 192.168.10.140:5432 D 24.181.10.1:49159	SNAT ⇨	S 12.181.128.1:24157 D 24.181.10.1:49159	⇨	Tuple match. (Firewall) S 12.181.128.1:24157 D 24.181.10.1:49159	DNAT ⇨	S 12.181.128.1:24157 D 10.0.0.150:5432 DUDP_ACK2=Stop shotgun.
T+ 1.0s							
Client A sends packets to punch hole (DUDP_ACK)	S 192.168.10.140:5432 D 24.181.10.1:49159	SNAT ⇨	S 12.181.128.1:24157 D 24.181.10.1:49159	⇨	S 12.181.128.1:24157 D 24.181.10.1:49159	DNAT ⇨	S 12.181.128.1:24157 D 10.0.0.150:5432 DUDP_ACK=use port #.
T+ 1.1s							
Client B sends ACK2 Packets (DUDP_ACK2)	S 24.181.10.1:49159 D 192.168.10.140:5432 DUDP_ACK2=Stop shotgun.	DNAT ⇦	S 24.181.10.1:49159 D 12.181.128.1:24157	⇦	S 24.181.10.1:49159 D 12.181.128.1:24157	SNAT ⇦	S 10.0.0.150:5432 D 12.181.128.1:24157
T+ 2.0s							
Client A sends RTP Packets	S 192.168.10.140:5432 D 24.181.10.1:49159	SNAT ⇨	S 12.181.128.1:24157 D 24.181.10.1:49159	⇨	S 12.181.128.1:24157 D 24.181.10.1:49159	DNAT ⇨	S 12.181.128.1:24157 D 10.0.0.150:5432
Client B sends RTP Packets	S 24.181.10.1:49159 D 192.168.10.140:5432	DNAT ⇦	S 24.181.10.1:49159 D 12.181.128.1:24157	⇦	S 24.181.10.1:49159 D 12.181.128.1:24157	SNAT ⇦	S 10.0.0.150:5432 D 12.181.128.1:24157

One shotgun packet gets through.
Match for known open sessions:

S 7.8.9.10:5432
D 12.181.128.1:24154 opened and responded

and

S 24.181.10.1:49159
D 12.181.128.1:24157 opened, and now responded

FW/Nat forwards packets:
Match for known open sessions:

S 7.8.9.10:5432
D 12.181.128.1:24154 opened and responded

S 24.181.10.1:49159
D 12.181.128.1:24157 opened, and now responded

66

Figure 6: Shotgun Exchange between Client behind Symmetric NAT/PATs, part 2 of 2.

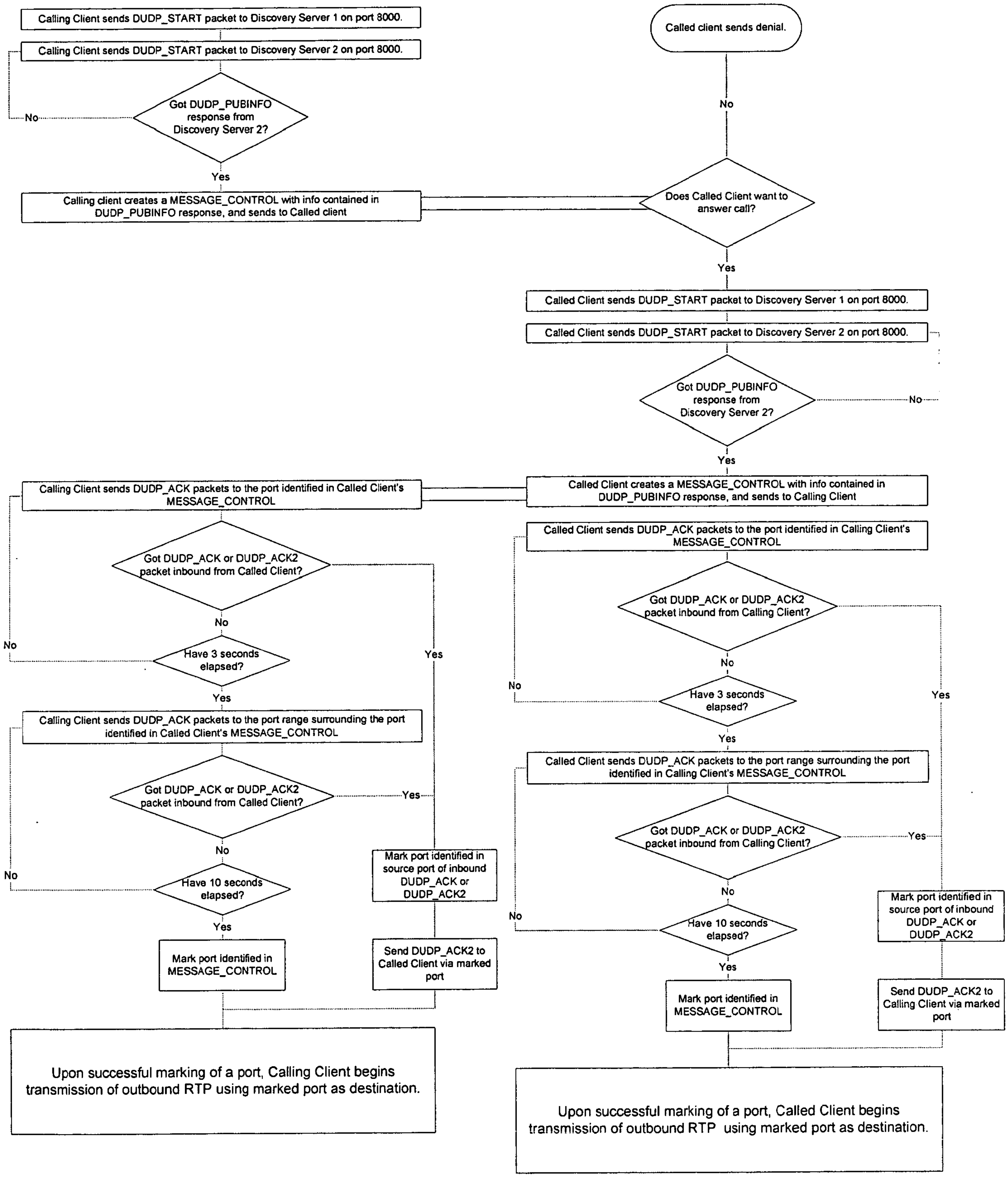

Figure 7: Flowchart of Discovery, Message Exchange, and Shotgun process

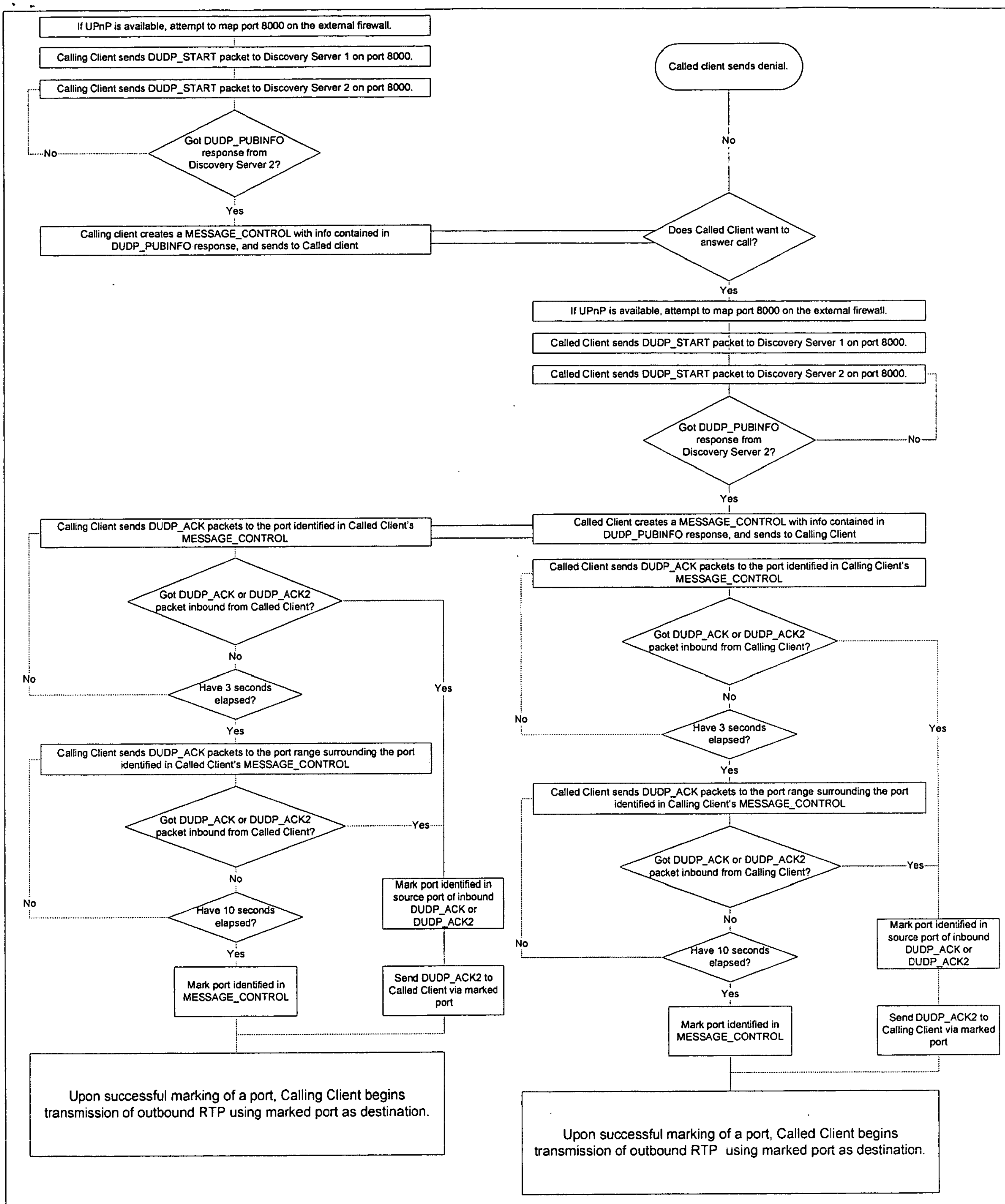

Figure 8: Flowchart of Discovery, Message Exchange, and Shotgun process, including UPnP

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) AND 1.27 (c)) - SMALL BUSINESS CONCERN			Docket No. 5636-104P
Serial No.	Filing Date	Patent No.	Issue Date

Applicant/ Patentee: **William L. Gaddy, Chang Feng, Timothy Michael Hingston, Chidambaram Ramanathan**

Invention: **SYSTEM AND METHOD FOR PEER-TO-PEER CONNECTION OF CLIENTS BEHIND SYMMETRIC FIREWALS**

I hereby declare that I am:

- ☐ the owner of the small business concern identified below:
- ☒ an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF CONCERN: **NetGen Video LLC**
ADDRESS OF CONCERN: **130 Campus Drive, Edison, NJ 08837**

I hereby declare that the above-identified small business concern qualifies as a small business concern as defined in 13 CFR 121.3-18, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees under Section 41(a) and (b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the above identified invention described in:

- ☒ the specification filed herewith with title as listed above.
- ☐ the application identified above.
- ☐ the patent identified above.

If the rights held by the above-identified small business concern are not exclusive, each individual, concern or organization having rights to the invention is listed on the next page and no rights to the invention are held by any person, other than the inventor, who could not qualify as an independent inventor under 37 CFR 1.9(c) or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

- ☒ no such person, concern or organization exists.
- ☐ each such person, concern or organization is listed below.

FULL NAME ____________________
ADDRESS ____________________
☐ Individual ☐ Small Business Concern ☐ Nonprofit Organization

FULL NAME ____________________
ADDRESS ____________________
☐ Individual ☐ Small Business Concern ☐ Nonprofit Organization

FULL NAME ____________________
ADDRESS ____________________
☐ Individual ☐ Small Business Concern ☐ Nonprofit Organization

FULL NAME ____________________
ADDRESS ____________________
☐ Individual ☐ Small Business Concern ☐ Nonprofit Organization

Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING: **Tor Dybfest**
TITLE OF PERSON SIGNING OTHER THAN OWNER: **Chief Financial Officer**
ADDRESS OF PERSON SIGNING: **4 Nelson Ridge Road, Princeton, NJ 08540**

SIGNATURE: [signature] DATE: 1/12/04